Making an automatic speech recognition service freely available on the web

Stuart N. Wrigley, Thomas Hain

Department of Computer Science, University of Sheffield
Regent Court, 211 Portobello, Sheffield S1 4DP, UK
s.wrigley@dcs.shef.ac.uk, t.hain@dcs.shef.ac.uk

Abstract

The state-of-the-art speech recognition system developed by the AMIDA project and which performed well in the NIST RT'09 evaluation has been made available as a web service. The service provides free access to ASR aimed specifically at the scientific community. There are two ways in which this service can be accessed: via a standard web-browser and programmatically via an API.

Index Terms: speech recognition, interface, api, transcription, web service

1. Introduction

Automatic speech recognition (ASR) is becoming a common feature of many products and applications ranging from smartphone utilities to automated call centres and beyond. However, the majority of such services tend to tailored to the specific needs of the customer; in this way, optimal performance can be gained by training and testing the system on a specific domain. The disadvantage of this approach is that (commercial) effort is largely focused on high return markets (professional dictation, etc); speech recognition systems for other fields, if available at all, are usually embedded in applications and not freely accessible.

Today, research and applications focussed on the processing of natural language texts are thriving; it is becoming increasingly common for these to be derived from previous unusable sources such as audio recordings. This activity is driving the necessity for easy to use, easy to access speech transcription services; however, most groups do not have access to research-grade speech recognition software.

In this paper we present a web-based interface[1] to our state-of-the-art speech recognition systems [1] derived from the AMIDA[2] NIST RT'09 systems [2]. The aim of this interface is to provide the non-commercial scientific research community with an interface to free speech transcription for domains and applications where the generation of such transcripts was not previously feasible. The service focusses on the provision of high quality offline, adaptive, speech-to-text for recordings of arbitrary length, usually incorporating multiple speakers and made in potentially challenging environments. The emphasis on offline recognition of long recordings is in contrast with other web-accessible speech services which target online, realtime spoken-dialog-like applications (e.g., [3]).

There are two distinct modes of access: via a browser or, programmatically, via an HTTP API. Both modes allow: the upload of audio data (the API also allows optional segmentation metadata); status checking for currently running ASR jobs; and the retrieval of a transcript. The browser-based interface provides a much richer experience by also allowing full control over existing uploads, ASR processing jobs and transcripts. The webASR service allows the transcript to be downloaded in a number of formats including MLF, PDF and HTML. Integration of the API-based service is facilitated by a wrapper DLL for Microsoft Windows platforms.

[1] http://www.webasr.org/

[2] http://www.amiproject.org

2. WebASR design

The goal of webASR is to provide free access to state-of-the-art speech recognition to as wide a community as possible (non-commercial) while at the same time having as low an adoption overhead as possible. To this end, it was decided that the core means of interacting with the service would be via a standard web browser; this removed the necessity for platform specific development, requires no software be installed on the user's computer and allows the service to be accessed anywhere in the world while still having access to all previous uploads and transcripts. The webASR service is implemented as a Java Servlet based web application hosted using the Apache Tomcat open source servlet container. Access to the system is restricted to registered (and manually approved) users, some of whom are administrators.

2.1. Browser interface

The primary mode of interaction with the service is via the browser-based interface.

2.1.1. User functionality

The first step to using the system is to register via the web form. This gathers basic information regarding the users name and affiliation. The registration request is logged by the system and an appropriate message is displayed when an administrator logs in. Once approved, a user can perform three high-level activities: manage their profile, manage their existing uploads and, finally, upload audio files containing speech to be transcribed.

Profile management. This allows the user to ensure their login and contact details are up to date on the system.

Audio upload. Before ASR processing can proceed, the audio containing the speech must be uploaded together with relevant metadata. This metadata covers two broad areas: information about the environment in which the recording took place and information about the speech content itself. The former captures details of the number of microphones used, the type of microphone (lapel, headset, farfield, array, etc.) and the physical location of the recording (street, office, etc.). Metadata regarding the speech content covers aspects such as the number of participants, the gender mix, the type of conversation (discussion, free conversation, presentation, etc.) and the topics that

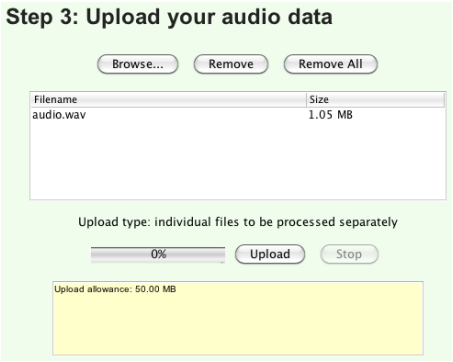

Figure 1: Java Applet to control upload of audio data.

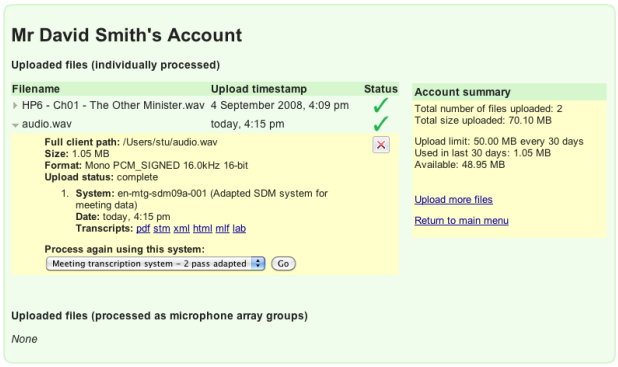

Figure 2: Account information showing uploads and available transcripts.

appeared in the discourse. The final stage is the selection of the audio files to be uploaded. This is achieved through the use of a Java Applet (Fig. 1). The applet approach allows the files to be validated before upload to ensure they are of a supported audio type and would not exceed the user's upload allowance. Once all the files have been selected the files are transferred to the webASR servers for processing.

Upload management. Once one or more audio files have been uploaded, they, together with their associated ASR processing and transcripts, can be managed using the interface shown in Fig. 2. The user's account page provides information regarding the status of their account as well as details of all the uploads that they have made. For the example user shown in Fig. 2, the right hand panel provides information about their account while the lefthand side of the panel lists all the uploads ordered by date. Each upload has an icon associated with it to quickly convey the processing status of the audio upload. In this case, both audio uploads have successfully completed their ASR processing (indicated by a green tick). For more information on each upload, the user is able to click on the grey arrow to the left of the filename to expose the *upload and transcript pane*. For each upload, aspects of the metadata are shown such as the path of the audio file (based on the user's filesystem not that of the webASR server), the audio file format and the file size.

The webASR service has been designed to allow each upload to be processed multiple times. Each time an ASR system is used to produce a transcript of an audio file, the details of this processing is shown as a numbered list with the *upload and transcript pane*. In this example, the audio file has been processed once by the *en-mtg-sdm09a-001 (Adapted SDM system for meeting data)* system. Indeed, as can be seen from the drop down box at the bottom of the *upload and transcript pane*, the user has the option of having this audio file processed with an alternative ASR system; in this case one entitled *Meeting transcription system - 2 pass adapted*. For each ASR process, a transcript is generated which, internally, is stored as XML. This allows the service to offer a wide range of output formats (PDF, STM, XML, HTML, MLF, etc.) with maximum efficiency by using 'on-the-fly' XSLT conversions.

2.2. Application programming interface (API)

An application programming interface (API) specifies the precise details of how a software program or service can be accessed by another software agent. In the context of the webASR service, the webASR API specifies the way in which any software program can upload audio and retrieve the associated transcripts via HTTP with no recourse to the browser-based interface described in Section 2.1. In many respects, the usage model of the API is a restricted version of the browser-based interface and follows the same underlying HTTP protocol. API-based functionality covers authentication, audio upload, transcript download as well as a number of utility functions such as determining the supported audio file formats and retrieving an estimate of when a transcript will be ready (see [4]).

Integration of the API-based service is facilitated by the availability of a wrapper DLL for Microsoft Windows platforms. This was written in C# and thus the source code can be integrated with any .NET application. The API service has also been integrated in native Mac OS X applications.

3. Conclusions

We have presented the webASR interface to the state-of-the-art AMIDA speech transcription system. This web service provides free access to research-grade ASR [2] for any non-commercial researcher wishing to transcribe audio recordings. A range of transcript formats are available to suite most needs; the service can also be accessed programmatically via an API thus allowing the service to be integrated into applications and other services.

4. Acknowledgements

The authors would like to thank all contributors to webASR speech recognition systems, from the AMI Consortium, or otherwise. In particular we would like to mention Vincent Wan and Asmaa El Hannani formerly of University of Sheffield; Lukas Burget and Martin Karafiat from University of Technology, Brno; Mike Lincoln from University of Edinburgh; John Dines and Phil Garner from Idiap; Thomas Niesler and Febe de Wet from University of Stellenbosch; Phil Woodland from Cambridge University; and many more contributors from these and other sites. This work was partly supported by the European IST Programme Project AMIDA (Augmented Multi-party Interaction with Distance Access) FP6-033812.

5. References

[1] T. Hain, A. el Hannani, S. Wrigley, and V. Wan, "Automatic speech recognition for scientific purposes - webASR," in *Interspeech'08*, 2008, pp. 504–507.

[2] T. Hain, L. Burget, J. Dines, P. N. Garner, A. El Hannani, M. j. Huijbregts, M. Karafiat, M. Lincoln, and V. Wan, "The AMIDA 2009 Meeting Transcription System," in *Interspeech'10*, 2010, pp. 358–361.

[3] G. Di Fabbrizio, T. Okken, and J. G. Wilpon, "A speech mashup framework for multimodal mobile services," in *ICMI-MLMI09*, 2009, pp. 71–78.

[4] S. N. Wrigley and T. Hain, "Web-based automatic speech recognition service - webasr," in *Interspeech 2011*, 2011.